

可视分析系统中的算法实践 —— 以电气仿真系统为例

作者：王琦

创建日期：2017 年 5 月 13 日

最后修改：2017 年 5 月 16 日

1. 异常检测中的区间生成问题

1.1 问题描述

我们需要通过不同物理量来判断电网的稳定性，每个物理量都会随时间动态变化。通过异常检测算法，我们可以求得许多异常时刻 t_1, t_2, \dots, t_n ，这些异常时刻满足 $t_1 < t_2 < \dots < t_n$ 。通过这些异常时刻，我们可以生成多个区间。区间 $[start, end)$ 表示存在连续的异常时刻

$start, start + 1, \dots, end - 1$

1.2 输入与输出

例如，有异常时刻 $1, 2, 5, 7, 8, 9$ 我们便可以生成异常区间集合

$\{[1, 3), [5, 6), [7, 10)\}$

1.3 练习

实现 `generate_intervals.py` 下的 `generate_intervals` 函数；

在 `test.py` 中设计测试用例，同时也将其他同学设计的测试用例添加进改文件；

在命令行下运行下述命令，检验代码的正确性。

```
python -m unittest
```

2. 不同检测指标下的区间合并问题（交集）

2.1 问题描述

我们可以通过不同的检测方法，来获得异常区间集合。比如：

- 采用算法 A（比如说 Shewhart 算法），我们可以获得一个区间集合 S_1

- 采用算法 B（比如说检测电压高于正常值的时刻），我们可以获得一个区间集合 S_2

我们希望求出 S_1 与 S_2 的交集，同时向用户展示满足条件 1 **或** 满足条件 2 的异常区间。

2.2 输入与输出

例如， $S_1 = \{[1, 3), [7, 10)\}$ ， $S_2 = \{[3, 4), [6, 9)\}$ 。 S_1 与 S_2 取交集的结果 $S = \{[1, 4), [7, 10)\}$ 。

2.3 练习

实现 `merge_intervals.py` 下的 `merge_intervals` 函数：

在 `test.py` 中设计测试用例，同时也将其他同学设计的测试用例添加进改文件：

在命令行下运行下述命令，检验代码的正确性。

```
python -m unittest
```

3. 最短路问题

电力网络由 **点**（发电站、变压器、母线等）和 **边**（交流线）组成。当某点 s 发生故障时，由故障所引发的冲击将沿着边，由近及远向其他点传播。为了分析其余点 v 是如何受到故障冲击的，计算由 s 到 v 的距离就显得很有意义。

3.1 Dijkstra 算法：求解单源最短路问题

下述文献给出了 Dijkstra 算法的描述：

- 《数据结构与算法分析:C语言描述》(Data Structures and Algorithm Analysis in C) 第 2 版 **9.3.2** 节
- [Dijkstra's algorithm - Wikipedia](#)
- 《算法导论》(Introduction to Algorithm, CLRS) 第 3 版 **24.3** 节（该章节也给出了证明，供学有余力的同学参考）

请参照算法描述，求解下述问题：

- [HackerRank. Dijkstra: Shortest Reach 2](#)

3.2 Floyd 算法：求解所有点对间的最短距离

假设我们现在想知道 **任意** 点 u, v 之间的最短距离，那么可以有如下两种解决方案：

- 以点 u 为源点，利用上文的 Dijkstra 算法，求出从 u 到 v 的最短距离
- 用弗洛伊德算法 (Floyd), 在一个二维数组中记录好任意点 u, v 间的距离，当每次发生查询时，查询数组即可。

下述文献给出了 Floyd 算法的描述：

- 《数据结构与算法分析:C语言描述》(Data Structures and Algorithm Analysis in C) 第 2 版 10.3.4 节
- [Floyd–Warshall algorithm - Wikipedia](#)
- 《算法导论》(Introduction to Algorithm, CLRS) 第 3 版 25.2 节（该章节也给出了证明，供学有余力的同学参考）

请参照算法描述，求解下述问题：

- [HackerRank. Floyd: City of Blinding Lights](#)